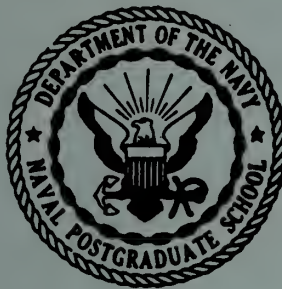


UNITED STATES NAVAL POSTGRADUATE SCHOOL



AN ITERATIVE SOLUTION
TO THE
GENERALIZED EIGENVALUE - EIGENVECTOR PROBLEM

by

R. D. Brunell

July 1966

TECHNICAL REPORT/RESEARCH PAPER NO. 69

Distribution of this document is unlimited

TA7
.U62
no.69

AN ITERATIVE SOLUTION
TO THE
GENERALIZED EIGENVALUE - EIGENVECTOR PROBLEM

Ronald by
R. D. Brunell
"

July 1966

TECHNICAL REPORT/RESEARCH PAPER NO. 69

Distribution of this document is unlimited

TR 7. 462 no 69

UNITED STATES NAVAL POSTGRADUATE SCHOOL

Monterey, California

Rear Admiral E. J. O'Donnell, USN,

Superintendent

R. F. Rinehart

Academic Dean

ABSTRACT:

This paper presents an approach to the solution of what is usually called "the generalized eigenvalue problem." The basic format of the method is similar to that presented by I. Tarnove (3), but with revisions in the root-finding and scaling procedures. The capability of calculating the associated eigenvectors has also been added to the previously-published algorithm.

This method uses an iterative root-finding technique and will converge with almost any first estimate. To preserve accuracy, the original matrix is used throughout for the evaluation of the function and scaling is done by powers of two.

A FORTRAN-63 program for the CDC-1604, using this method, has been written for matrices having polynomial elements. The code was evaluated using a number of trial problems, including pathological matrices, and some of these results have been incorporated into this report.

Prepared by: R. D. Brunell

Mathematician, Computer Facility

Approved by:

D. G. Williams

Head, Computer Facility

Released by:

C. E. Menneken

Dean of

Research Administration

U. S. Naval Postgraduate School Technical Report/Research Paper No. 69

July 1966

UNCLASSIFIED

TABLE OF CONTENTS

Title Page	1
I. Statement of the Problem	2
II. Method of Solution	3
III. General Discussion	5
IV. Numerical Examples	6
V. Areas of Further Investigation	15
References	16
Appendix I: Flow Chart	17
Appendix II: Subroutine Listings	20
GENEIG	20
CALNEWX	23
CDTERM	25
JORCOM	27
CALFUNC	29
CKEVEC	30
FUNCEV	31
ROMAT	32
ACCEPT	33
SCALE2	34
SCALE1	36
EVEK	37
Appendix III: Subroutine Description	41
<u>F4-NPGS-GENEIG, A Method for the Solution of the</u> <u>"Generalized Eigenvalue Problem" with Polynomial</u> <u>Elements, June 1966.</u>	

I. Statement of the Problem

The "generalized eigenvalue problem" may be stated as follows:

Find those values of the complex parameter, z , which will provide a non-trivial solution to the system of linear equations,

$$\begin{aligned}
 &A_{11}(z)X_1 + A_{12}(z)X_2 + \dots + A_{1n}(z)X_n = 0 \\
 &A_{21}(z)X_1 + A_{22}(z)X_2 + \dots + A_{2n}(z)X_n = 0 \\
 &\quad \cdot \quad \quad \quad \cdot \quad \quad \quad \cdot \\
 (1) \quad &\quad \cdot \quad \quad \quad \cdot \quad \quad \quad \cdot \\
 &\quad \cdot \quad \quad \quad \cdot \quad \quad \quad \cdot \\
 &A_{n1}(z)X_1 + A_{n2}(z)X_2 + \dots + A_{nn}(z)X_n = 0.
 \end{aligned}$$

This can be written more concisely as

$$(2) \quad A(z)X = 0$$

where A is an $n \times n$ matrix, the elements of which are arbitrary complex functions of z , and X is a complex column vector of n elements. The set $\{z_i : A(z_i)X_i = 0\}$ is the set of eigenvalues and the X_i are called the associated eigenvectors.

The special case where the $A_{ij}(z)$'s are polynomials in z of degree 2 or less is encountered most often in practical work. The rest of the discussion in this paper will be confined to problems of this type. However, it should be pointed out that the general method described here is not so restricted.

Under the above restriction we can rewrite $A(z)$ as

$$(3) \quad A(z) = A_0 z^2 + A_1 z + A_2$$

where the A_i 's are constant complex matrices. We will need this form in the later discussion on scaling.

II. Method of Solution

The heart of the method described here is the iterative root-finding technique. After a literature search, in which much background information was gained from Wilkinson (1), and some experimentation, it was decided that a method developed by J. F. Traub (2) would be used in this code. Muller's method (7) has been mentioned by a number of authors in connection with iterative solutions in the complex plane; however, Traub's technique was chosen because of its simplicity and ease of computation. The methods are similar; both are of second order and have the same asymptotic error constant. The main difference is that Traub uses the Newtonian form of the interpolating polynomial and Muller uses the Lagrangian form.

In essence the technique is to perform a quadratic fit using the last three iterants and their functional values. This algorithm is defined as follows:

$$(4) \quad z_{i+1} = z_i - 2f_i / (w \pm \{w^2 - 4f_i f[z_i, z_{i-1}, z_{i-2}]\}^{1/2})$$

where

$$(5) \quad w = f[z_i, z_{i-1}] + (z_i - z_{i-1})f[z_i, z_{i-1}, z_{i-2}]$$

and

$$(6) \quad f[z_i, z_{i-1}, z_{i-2}] = (f[z_i, z_{i-1}] - f[z_{i-1}, z_{i-2}]) / (z_i - z_{i-2})$$

The variable sign in equation (4) is chosen so that the denominator has the larger magnitude.

$$f(z) = A_0 z^2 + A_1 z + A_2$$

Convergence is determined by either of the two tests,

$$(7) \quad |z_i - z_{i-1}| / |z_i| < e_1$$

or

$$(8) \quad |f_i| < e_2 .$$

If either of these inequalities is met the process is considered to have converged to a root.

In order to avoid convergence to a previously-calculated root, a method of suppressing zeros, first suggested by G. E. Forsythe (Reference 3, page 164) was incorporated into the code. The first root, z_1 , is found as a zero of $f(z)$ and the remaining roots, z_i , are found as zeros of $g_i(z)$, where

$$(9) \quad g_i(z) \equiv f(z) / \prod_{k=1}^i (z - z_k) , \quad i > 1 .$$

One of the problems encountered in a program of this type is floating-point exponential overflow when working with matrices of large order. To alleviate this problem an automatic scaling feature was devised. This scaling is done in two parts, a priori and a posteriori. The a priori scaling is done at the beginning of the problem and is carried out in the following manner. The coefficients of each polynomial element in a row of the matrix A are summed and a maximum is found. In a binary computer this maximum may be expressed exactly as $bx2^j$. Then each coefficient in every polynomial in that row is divided by 2^j . This operation is carried out for each row in the matrix.

A posteriori scaling is done only when an overflow occurs. If, at the time of the overflow, the estimate to the root being calculated is some number, say $cx2^k$, then $A(z)$ (defined in equation (3)) is replaced by

$$(10) \quad A'(z') = A_0 z'^2 + \frac{A_1}{2^k} z' + \frac{A_2}{(2^k)^2} ,$$

and a root, z' , is then found for the new problem. The root of the original problem is then calculated as

$$(11) \quad z_i = 2^k z'$$

The scaled problem is started by taking the last three iterants in the original problem and dividing them by 2^k . After a root is found the matrix is returned to its original form and the program proceeds to find the next root.

In many cases it is desirable to obtain the eigenvectors associated with the eigenvalues. An option to perform these calculations is available in the program. Because there exists no procedure which will absolutely assure the finding of an eigenvector, the code will make $2n$ attempts, where n is the order of the matrix, at computing an eigenvector for each eigenvalue. This procedure is carried out as outlined below.

An assumption is made that the elements of the vector may be written as multiples of a single element in the vector. One possible representation of this is

$$(12) \quad X_i^T = (x_{1i}, b_1 x_{2i}, b_2 x_{3i}, \dots, b_{n-1} x_{ni}) .$$

Using this assumption the code makes its first attempt by setting $X_{1j}=1$ and solving the system of equations (13):

$$(13) \quad \begin{pmatrix} a_{22} & a_{23} & \dots & a_{2n} \\ a_{32} & a_{33} & \dots & a_{3n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n2} & a_{n3} & \dots & a_{nn} \end{pmatrix} \begin{pmatrix} X_{2i} \\ X_{3i} \\ \vdots \\ X_{ni} \end{pmatrix} = \begin{pmatrix} -a_{21} \\ -a_{31} \\ \vdots \\ -a_{n1} \end{pmatrix} ,$$

where a_{ij} is the function A_{ij} evaluated at z_i .

The expression, $A(z) X$, is then calculated using this vector and the result compared to the zero vector. If the vector, X , is found to be unacceptable a second attempt is made by replacing a_{2k} by a_{1k} , $k=1,2,\dots,n$, in (13). If this fails then x_{2i} is set equal to unity and the system (14) is solved,

$$(14) \quad \begin{pmatrix} a_{11} & a_{13} & \dots & a_{1n} \\ & a_{31} & a_{33} & \dots & a_{3n} \\ & \cdot & \cdot & & \cdot \\ & \cdot & \cdot & & \cdot \\ a_{n1} & a_{n3} & \dots & a_{nn} \end{pmatrix} \begin{pmatrix} X_{1i} \\ X_{3i} \\ \cdot \\ \cdot \\ X_{ni} \end{pmatrix} = \begin{pmatrix} -a_{12} \\ -a_{32} \\ \cdot \\ \cdot \\ -a_{n2} \end{pmatrix}$$

and checked as indicated above. A fourth attempt may be made by replacing, in (14), a_{3k} by a_{2k} for $k=1,2,\dots,n$. The process continues for all X_{ji} or until an eigenvector is found. If no eigenvector is found a statement to that effect is printed along with the results of the final attempt.

III. General Discussion of Method

A code, using the method described in this paper, has been written in FORTRAN-63 for use on a CDC 1604. This program has been checked out on a variety of problems, several of which are shown in the Section IV. The accuracy of the process has been shown to be quite good. This may be attributed to several factors. First, only the original input matrix is used for the evaluation of the function each time, and, second, all scaling is done by powers of two (Reference (8), page 98). The method has been shown to be reasonably fast, although a good relative comparison is impossible because the number of iterations required is a function of

the estimate provided. Unfortunately this information generally has not been included in previous literature on this subject. Because the number of computations necessary to determine the next estimate to the root is so small compared to those required to evaluate the determinant, the time required for each iteration can be considered substantially equal to the time necessary for the latter. The speed of the program is further enhanced by the fact that through the use of good estimates only eigenvalues of interest need be computed, although convergence seems to result from almost arbitrary estimates. Also because of the feature which forces avoidance of previously found zeros, it is possible to restart the program and by-pass any such prior calculations.

IV. Numerical Examples

Most of the examples used for this section have been taken from papers on the standard eigenvalue problem. This was necessary because of the lack of comparative data on the generalized problem. However, these examples should be adequate to demonstrate the ability of this method to find pathological roots and also to get some comparison as to its accuracy. All problems, using the method described in this report, were run on a CDC 1604 which has a 48-bit word length.

Example 1

(See Parlett (5)). These results were obtained using Laguerre's method on an IBM 7090 which has a 36-bit word. A is defined as

611-z	196	-192	407	-8	-52	-49	29
196	899-z	113	-192	-71	-43	-8	-44
-192	113	899-z	196	61	49	8	52
407	-192	196	611-z	8	44	59	-23
-8	-71	61	8	411-z	-599	208	208
-52	-43	49	44	-599	411-z	208	208
-49	-8	8	59	208	208	99-z	-911
29	-44	52	-23	208	208	-911	99-z

Table of Eigenvalues

Actual (8 digits)	Parlett (5)	Present Method
1020.0490	1020.0500	1020.0490
1020.0000	1019.9997	1020.0000
1019.9020	1019.9019	1019.9019
1000.0000	1000.0001	1000.0000
1000.0000	999.99999	1000.0000
.098048640	.098045509	0.098048646
0.0	-0.00000094	-0.000000003
-1020.0490	-1020.0490	-1020.0490

Total running time: 16 seconds

Average number of iterations per eigenvalue: 10.5

Example 2

(See Barlow and Jones (4)). Their work was done on an IBM 7040 (36-bit word) using an extension of the classical secant method to the complex field. This paper (4) also gives results obtained by Eberlein (6) using a Jacobi-like method.

$$A = \begin{pmatrix} 15-z & 11 & 6 & -9 & -15 \\ 1 & 3-z & 9 & -3 & -8 \\ 7 & 6 & 6-z & -3 & -11 \\ 7 & 7 & 5 & -3-z & -11 \\ 17 & 12 & 5 & -10 & -16-z \end{pmatrix}$$

Table of Eigenvalues

Actual (6)	Eberlein (6)	Barlow & Jones (4)	Present Method
-1.0	-.999	-1.0000004	-.9999999999
1.50016 + 3.57064i	1.505 + 3.57i	1.49850 + 3.57027i	1.50001 + 3.57073i
1.50016 + 3.57064i	1.495 + 3.57i	1.50150 + 3.57110i	1.50003 + 3.57069i
1.50016 - 3.57064i	1.505 - 3.57i	1.49996 - 3.57127i	1.50001 - 3.57074i
1.50016 - 3.57064i	1.495 - 3.57i	1.49864 - 3.57023i	1.49999 - 3.57070i

Total running time: 4.38 seconds

Average number of iterations per eigenvalue: 6

Example 3

(See Parlett (5)). This example was used to demonstrate the ability of the code to extract only a few desired roots.

$$A = (B - zI)$$
$$B = \begin{pmatrix} C & 2C \\ 4C & 3C \end{pmatrix} \quad C = \begin{pmatrix} 5D - D & D \\ 5D & D \end{pmatrix} \quad D = \begin{pmatrix} -2 & 2 & 2 & 2 \\ -3 & 3 & 2 & 2 \\ -2 & 0 & 4 & 2 \\ -1 & 0 & 0 & 5 \end{pmatrix}$$

The true eigenvalues of this system (Eberlein (6)) are $60 \pm 20i$, $45 \pm 15i$, $30 \pm 10i$, $15 \pm 5i$, $-12 \pm 4i$, $-9 \pm 3i$, $-6 \pm 2i$, $-3 \pm i$.

Initial Guess	Eigenvalue Found
-2.5 - 0.625i	-3.00000006 - 9.9999947i
-2.5 + 0.625i	-3.00000005 + 1.0000000i
-5.0 - 1.25i	-6.00000055 - 1.9999948i
-5.0 + 1.25i	-5.9999975 + 2.0000003i

Total running time: 55.28 seconds

Average number of iterations per eigenvalue: 17.5

Example 4

In this example both the eigenvalues and eigenvectors were found.

$$A = \begin{pmatrix} 1-z & -2 & 3 & -2 \\ 1 & 5-z & -1 & -1 \\ 2 & 3 & 2-z & -2 \\ 2 & -2 & 6 & -3-z \end{pmatrix}$$

Table of Eigenvalues and Eigenvectors

True Values		Computed Values	
Eigenvalues	Eigenvectors	Eigenvalues	Eigenvectors
-1	1 0 0 1	-1.00000000	1.0 0.0 0.0 1.0
2	1 -6 -9 -8	1.999939	1.0 -5.9993 -8.9991 -7.9993
2	1 -6 -9 -8	2.00007	1.0 -6.0007 -9.001 -8.0008
2	1 -6 -9 -8	1.99998	1.0 -5.9998 -8.9998 -7.9998

Total running time: 3.83 seconds

Average number of iterations per eigenvalue: 8.25

Example 5

This is an example of the generalized problem.

$$A = \begin{pmatrix} 2z^2+14z-88 & -z^2-13z-22 & z^2+2z-99 \\ -3z^2+21z-36 & 6z^2-6z-36 & 4z^2-48z+108 \\ 8z^2+8z-160 & z^2+7z+10 & -z^2+4z+45 \end{pmatrix}$$

Table of Eigenvalues and Eigenvectors

True Values		Computed Values	
Eigenvalues	Eigenvectors	Eigenvalues	Eigenvectors
-11	1 -29/6 153/40	-10.99999999	1.0 -4.833333333 3.82500000
-5	10/33 1 -27/154	-5.0	0.3030303030 1.0 -1.753246753
-2	0 1 0	-2.0	0.0 1.0 0.0
3	0 1 -5/6	3.0	-0.0000000001 1.0 -0.8333333333
4	1 0 0	4.0	1.0 0.0 0.0
9	0 0 1	9.0	0.0 0.0 1.0

Total running time: 3.85 seconds

Average number of iterations per eigenvalue: 4.0

Example 6

This final example of the generalized problem is constructed so that it can also be solved as a standard problem. It is defined as follows,

$$\text{i.) } A \equiv (B+zC) = 0,$$

where B and C are 12x12 matrices with constant elements. By multiplying on the left by C^{-1} a transformation is made to the standard problem,

$$\text{ii.) } A' \equiv (BC^{-1}+zI) = 0.$$

This problem has been solved in both forms by the technique discussed here.

In addition, it was solved using the algorithm in (5), so that the results might be compared.

The matrices B and C are defined on the following page.

The matrices B and C were defined as:

$$\begin{pmatrix}
 10 & -15 & 1 & -2 & -81 & 91 & -69 & 14 & 62 & -36 & -20 & 99 \\
 22 & -46 & -25 & -85 & -3 & 89 & 27 & 53 & -93 & -34 & 52 & 19 \\
 24 & 48 & -22 & -97 & -76 & -64 & -15 & 24 & 49 & -32 & 30 & -19 \\
 -42 & -93 & -6 & 61 & 7 & 16 & 39 & -53 & -71 & 57 & 0 & -74 \\
 37 & -39 & -81 & 16 & -6 & 91 & 60 & -81 & 49 & 60 & 14 & -6 \\
 -77 & -6 & 11 & -42 & 27 & 53 & 18 & -70 & -90 & -15 & 21 & -81 \\
 99 & -72 & 56 & 69 & 98 & 31 & 71 & 18 & -44 & 48 & -63 & -21 \\
 -96 & -91 & -5 & 7 & 18 & 20 & -94 & -56 & 69 & -60 & -18 & -84 \\
 -89 & 14 & -63 & -10 & -17 & -18 & 57 & 84 & -25 & 12 & 58 & -44 \\
 -85 & -36 & 53 & 53 & 53 & -59 & -38 & 62 & 8 & -17 & -16 & 11 \\
 28 & 69 & -88 & 33 & -70 & 79 & -56 & -5 & 90 & -31 & -1 & 85 \\
 -63 & -40 & -48 & -3 & 49 & -69 & -18 & -72 & 52 & -20 & 12 & -90
 \end{pmatrix} = B$$

$$\begin{pmatrix}
 -81 & -72 & -4 & -96 & 24 & -82 & 66 & 14 & -76 & 14 & 13 & 87 \\
 29 & -20 & 68 & 26 & -46 & -20 & 89 & 81 & -86 & -12 & -92 & 57 \\
 0 & 57 & 39 & 66 & -84 & -40 & 32 & 61 & -98 & -96 & 64 & 64 \\
 5 & -4 & -25 & 26 & -44 & 44 & -37 & 63 & 45 & 66 & 75 & 66 \\
 -91 & 26 & -64 & -94 & 26 & 25 & 39 & -22 & 71 & 64 & 91 & 42 \\
 0 & -4 & -87 & -77 & 42 & 35 & -74 & -99 & -81 & -42 & 43 & -76 \\
 0 & 69 & -62 & 56 & 86 & 88 & 76 & 36 & -84 & -93 & 76 & -65 \\
 -69 & -65 & 95 & -55 & 18 & 27 & -26 & -8 & -40 & 59 & -29 & 80 \\
 25 & 57 & 29 & 88 & -67 & 48 & 18 & -82 & 65 & 69 & -33 & 54 \\
 -9 & -83 & -73 & 12 & -30 & -18 & 28 & -35 & 5 & 41 & 34 & 37 \\
 -91 & -42 & 27 & 30 & 4 & -86 & 29 & 99 & 55 & -84 & 29 & 9 \\
 -17 & -56 & -90 & -49 & 20 & -59 & -6 & 20 & -18 & 2 & 73 & -83
 \end{pmatrix} = C$$

These three cases were run with a relative error bound of 10^{-6} and the results from all of them were identical to six or seven significant figures. The eigenvalues found are listed in the table below.

Table of Eigenvalues
.956212
-2.67567
- .0213507
- .466901
- .626829 \pm 2.56009
- .274574 \pm .327271
.254714 \pm .457145
- .796302 \pm 1.04215

Total time for present method: 104 seconds
Average number of iterations per eigenvalue: 17.2

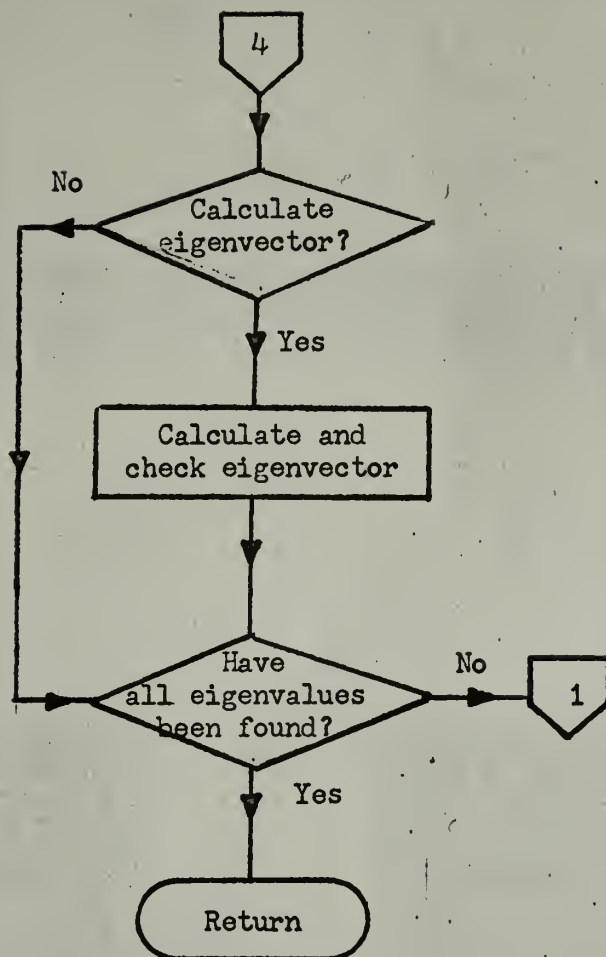
V. Areas of Further Investigation

Some additional features are being considered for inclusion in the present code. However, at the time of this report all problems involving these revisions had not been resolved.

It is felt that some process which could quickly determine the approximate location of eigenvalues may be useful, especially when large order problems are undertaken. It has not, as yet, been determined what method might be most profitable for this purpose.

An automatic entry into double precision would be desirable for parts of the computations. But it will be necessary to devise some general test to indicate when this move could best be made without wasting computer time.

A test to find when the domain of indeterminacy has been reached in the search for a root would be a valuable addition to those existing tests for acceptance of a root.




```

SUBROUTINE GENEIG
C
C      THIS IS THE CONTROL SUBROUTINE
C      THE FOLLOWING INFORMATION MUST BE SETUP IN COMMON FOR GENEIG
C
C      INPUT INFORMATION
C      THE ARRAY A MUST CONTAIN THE COEFFICIENT MATRICES
C      THE ARRAY EIGG MUST CONTAIN THE FIRST GUESSES
C      NN=RANK OF MATRICES
C      IORD=ORDER OF POLYNOMIAL ELEMENTS
C      IE=NUMBER OF FIRST EIGENVALUE TO BE FOUND
C      NEIG=NUMBER OF LAST EIGENVALUE TO BE FOUND
C      IMAX=MAXIMUM NUMBER OF ITERATIONS TO BE ALLOWED
C      ISAC=FLAG FOR APRIORI SCALING, ISAC=0 SCALING WILL OCCUR
C      IPRINT=FLAG FOR PRINT AT EVERY ITERATION,IPRINT=0 NO PRINTING
C      IVEC=FLAG FOR COMPUTING EIGENVECTORS,IVEC=0 NO EIGENVECTORS
C      EP1=RELATIVE ERROR BOUND ON EIGENVALUE
C      EP2=ABSOLUTE ERROR BOUND ON FUNCTION
C      EP3=RELATIVE ERROR BOUND ON EIGENVECTOR
C
C      OUTPUT INFORMATION
C      THE ARRAY EVAL WILL CONTAIN THE COMPUTED EIGENVALUES
C      THE ARRAY EVEC WILL CONTAIN THE ASSOCIATED EIGENVECTORS
C
C      DIMENSION AE(20,20)
C      COMMON A(20,20,3),EIGG(40,2),EVAL(40),EVEC(20,40),NN,IORD,IE,NEIG,
1IMAX,ISAC,IPRINT,IVEC,EP1,EP2,EP3
C      COMMON /GBLOC2/AE,XI,XI1,XI2,FI,F11,FI2,FX0,HI2,YXI1,YXI2,XS2,II
1,IJ,IJ,KJ,CC1,CC2,CC3,CC4,CC5,DD1,DD2,DD3,DD4,DD5,DD6,DD7,DD8
C      TYPE COMPLEX AE,XI,XI1,XI2,FI,F11,FI2,HI2,CC1,CC2,CC3,CC4,CC5
C      TYPE COMPLEX A,EIGG,EVAL,EVEC
C      XS2=1.0
C      DD8=0.0
C      WRITE (51,30)
30  FORMAT(1H1)
C      IF(ISAC) 20,10,20
C
C      APRIORI SCALING IF DESIRED
C
C      10 CALL SCALE1
C      20 DO 1000 IJ=IE,NEIG
C      DD1=0.0
C      II=1
C
000000
000010
000020
000030
000040
000050
000060
000070
000080
000090
000100
000110
000120
000130
000140
000150
000160
000170
000180
000190
000200
000210
000220
000230
000240
000250
000260
000270
000280
000290
000300
000310
000320
000330
000340
000350
000360
000370
000380
000390
000400
000410
000420

```


000860
000870
000880
000890
000900
000910
000920
000930
000940
000950
000960
000970
000980
000990

```
900 EVAL(IJ)=XI*XS2
    IF(DD1) 930,940,930
C
C      INVERSE TRANSFORMATION IF APOSTERIORI SCALING HAD OCCURED
C
930 CALL ROMAT
940 IF (IVEC) 950,1000,950
C
C      COMPUTE EIGENVECTORS IF DESIRED
C
950 CALL EVEK
1000 CONTINUE
    RETURN
    END
```



```

SUBROUTINE CALNEWX
C
C      THIS SUBROUTINE DETERMINES THE VALUE OF THE NEXT ITERANT
C      USING TRAUBS METHOD
C
      DIMENSION AE(20,20)
      COMMON A(20,20,3),EIGG(40,2),EVAL(40),EVEC(20,40),NN,IORD,IE,NEIG,
      1IMAX,ISAC,IPRINT,IVEC,EP1,EP2,EP3
      COMMON /GBLOC2/AE,XI,XI1,XI2,FI,F11,FI2,FX0,HI2,YXI1,YXI2,XS2,II
      1,IJ,IJ,KJ,CC1,CC2,CC3,CC4,CC5,DD1,DD2,DD3,DD4,DD5,DD6,DD7,DD8
      TYPE COMPLEX A,EIGG,EVAL,EVEC
      TYPE COMPLEX AE,XI,XI1,XI2,FI,F11,FI2,HI2,CC1,CC2,CC3,CC4,CC5
      TYPE COMPLEX HI1,HI2,DELX1,DELX2,W,T,XC,X,XX1,XX2
      IF(II-1) 10,10,50

      FIND INITIAL VALUES
C
C
C
      10 CALL FUNCEV(XI2,FI2)
      IF (IPRINT) 35,15,35
      35 WRITE (51,210) ,XI2,CC5
      210 FORMAT(/6X,12H1ST GUESS      ,2HX=,C(E22.11,E22.11),7H F(X)=,
      1C(E22.11,E22.11))
      15 CALL FUNCEV(XI1,FI1)
      IF (IPRINT) 25,45,25
      25 WRITE (51,220) ,XI1,CC5
      220 FORMAT(6X,12H2ND GUESS      ,2HX=,C(E22.11,E22.11),7H F(X)=,
      1C(E22.11,E22.11))
      45 CALL FUNCEV(XI,FI)
      IF (IPRINT) 55,65,55
      55 WRITE (51,230) ,XI,CC5
      230 FORMAT(6X,12H3RD GUESS      ,2HX=,C(E22.11,E22.11),7H F(X)=,
      1C(E22.11,E22.11))
      65 IF(DD8) 40,40,20

      DETERMINE IF APOSTERIORI SCALING IS NECESSARY
C
C
C
      20 CALL SCALE2
      DD8=0.0
      40 HI2=(FI1-FI2)/(XI1-XI2)

      COMPUTE NEXT ITERANT
C
C
C
      50 DELX1=XI-XI1

```


001430
001440
001450
001460
001470
001480
001490
001500
001510
001520
001530
001540
001550
001560
001570
001580
001590
001600
001610
001620
001630
001640
001650
001660
001670
001680

```

DELX2=XI-XI2
HI1=(FI-FI1)/DELX1
HI12=(HI1-HI2)/DELX2
W=HI1+DELX1*HI12
XC=CSQRT(W*W-4.0*FI*HI12)
XX1=W+XC
XX2=W-XC
XC1=CABS(XX1)
XC2=CABS(XX2)

      DETERMINE WHICH ROOT IS TO BE USED

      IF (XC1-XC2) 90,90,100
90  T=XI-(2.0*FI)/XX2
    GO TO 110
100 T=XI-(2.0*FI)/XX1

      SET VALUES FOR NEXT ITERATION

110 XI2=XI1
    XI1=XI
    XI=T
    FI=FI
    HI2=HI1
190 RETURN
    END

```



```

C      SUBROUTINE CDTERM (N,B,DET,KER,M)
C
C      THIS SUBROUTINE CALCULATES THE DETERMINANT USING THE
C      GAUSS METHOD WITH PIVOTING
C
C      DIMENSION B(2,20,20),DET(2)
C      C=1.0
C      DO 35 L=1,N
C      KP=0
C      Z=0.0
C      DO 12 K=L,N
C
C          FIND MAXIMUM ELEMENT IN COLUMN
C
C          BMAG=B(1,K,L)**2+B(2,K,L)**2
C          IF(Z-BMAG) 11,12,12
C      11 Z=BMAG
C      KP=K
C      12 CONTINUE
C          IF(L-KP)13,30,30
C
C          SIGN CHANGE IF PIVOTING IF NECESSARY
C
C      13 C=-C
C      DO 14 J=L,N
C
C          INTERCHANGE ROWS
C
C          TEMP=B(1,L,J)
C          Z=B(2,L,J)
C          B(1,L,J)=B(1,KP,J)
C          B(2,L,J)=B(2,KP,J)
C          B(1,KP,J)=TEMP
C          B(2,KP,J)=Z
C      14 CONTINUE
C      30 IF(L=N) 31,40,40
C      31 LP1=L+1
C      DO 35 K=LP1,N
C
C          GAUSS METHOD
C
C          IF(B(1,K,L)) 33,46,33
C      46 IF(B(2,K,L)) 33,35,33

```



```

33 Z=B(1,L,L)**2+B(2,L,L)**2
   IF(Z) 43,44,43
43 TEMP=(B(1,K,L)*B(1,L,L)+B(2,K,L)*B(2,L,L))/Z
   BMAG=(B(1,L,L)*B(2,K,L)-B(1,K,L)*B(2,L,L))/Z
   DO 34 J=LPI,N
     B(1,K,J)=B(1,K,J)-(TEMP*B(1,L,J)-BMAG*B(2,L,J))
     B(2,K,J)=B(2,K,J)-(TEMP*B(2,L,J)+BMAG*B(1,L,J))
34 CONTINUE
35 CONTINUE
40 DET(1)=B(1,1,1)
   DET(2)=B(2,1,1)
   DO 41 K=2,N
     BMAG=(DET(1)*B(1,K,K)-DET(2)*B(2,K,K))
     DET(2)=(DET(1)*B(2,K,K)+DET(2)*B(1,K,K))
     DET(1)=BMAG
41 CONTINUE
   IF (C) 50,44,60
50 DET(1)=-DET(1)
   DET(2)=-DET(2)
60 KER=1
   RETURN
44 KER=2
   RETURN
   END
002120
002130
002140
002150
002160
002170
002180
002190
002200
002210
002220
002230
002240
002250
002260
002270
002280
002290
002300
002310
002320
002330
002340
002350

```



```

C      SUBROUTINE JORCOM (A,N,X,KR,KC)
C
C      THIS SUBROUTINE COMPUTES THE EIGENVECTOR USING THE
C      GAUSS-JORDAN METHOD
C
C      DIMENSION A(KR,KC),X(KR)
C      TYPE COMPLEX A,X,XY
C
C      K=N+1
C      11 IF(K-1) 13,6,15
C      15 D=0.0
C
C      FIND MAXIMUM ELEMENT IN COLUMN
C
C      DO 2 I=2,K
C      II = I-1
C      C1=CABS(A(II,1))
C      IF(C1-D) 2,2,3
C      3 L=II
C      D=C1
C      2 CONTINUE
C      4 IF(L-1) 5,6,5
C
C      INTERCHANGE ROWS
C
C      5 DO 7 J=1,K
C      XY=A(L,J)
C      A(L,J)=A(1,J)
C      A(1,J)=XY
C      7 CONTINUE
C
C      SETUP SOLUTION VECTOR
C
C      6 DO 8 I=1,N
C      X(I)=A(I,1)
C      8 CONTINUE
C      IF (K-1) 12,13,12
C
C      GAUSS-JORDAN METHOD
C
C      12 DO 10 J=2,K
C      IJ=J-1
C      XY=A(1,J)/X(1)

```



```
DO 9 I=2,N
  A(I-1,IJ)=A(I,J)-X(I)*XY
  9 CONTINUE
  A(N,IJ)=XY
  10 CONTINUE
  K=K-1
  GO TO 11
  13 RETURN
  END
```

```
002790
002800
002810
002820
002830
002840
002850
002860
002870
```



```

SUBROUTINE CALFUNC(AZ)
C
C
C
      THIS SUBROUTINE EVALUATES THE POLYNOMIAL ELEMENTS AT XI
C
      DIMENSION AE(20,20)
      COMMON A(20,20,3),EIGG(40,2),EVAL(40),EVEC(20,40),NN,IORD,IE,NEIG,
      IIMAX,ISAC,IPRINT,IVEC,EP1,EP2,EP3
      COMMON /GBLOC2/AE,XI,XI1,XI2,FI,FI1,FI2,FX0,HI2,YXI1,YXI2,XS2,II
      1,IJ,IkJ ,CC1,CC2,CC3,CC4,CC5,DD1,DD2,DD3,DD4,DD5,DD6,DD7,DD8
      TYPE COMPLEX AE,XI,XI1,XI2,FI,FI1,FI2,HI2 ,CC1,CC2,CC3,CC4,CC5
      TYPE COMPLEX A,EIGG,EVAL,EVEC
      TYPE COMPLEX AZ
C
      IF(IORD-1) 10,20,50
      10 WRITE (51,15) ,IORD
      15 FORMAT(/4X,40HIMPOSSIBLE ORDER OF MATRIX ELEMENTS      =,I10 )
      STOP
      20 DO 40 I=1,NN
      DO 30 J=1,NN
      AE(I,J)=A(I,J,2)*AZ+A(I,J,1)
      30 CONTINUE
      40 CONTINUE
      GO TO 80
      50 DO 70 I=1,NN
      DO 60 J=1,NN
      AE(I,J)=AZ*(A(I,J,3)*AZ+A(I,J,2))+A(I,J,1)
      60 CONTINUE
      70 CONTINUE
      80 RETURN
      END
C
002880
002890
002900
002910
002920
002930
002940
002950
002960
002970
002980
002990
003000
003010
003020
003030
003040
003050
003060
003070
003080
003090
003100
003110
003120
003130
003140
003150
003160
003170

```



```

SUBROUTINE FUNCEV(AZ,BZ)
C
C      THIS SUBROUTINE EVALUATES FUNCTION AT PRESENT ITERATE VALUE
C
      DIMENSION AE(20,20)
      COMMON A(20,20,3),EIGG(40,2),EVAL(40),EVEC(20,40),NN,IORD,IE,NEIG,
      1IMAX,ISAC,IPRINT,IVEC,EP1,EP2,EP3
      COMMON /GBLOC2/AE,XI,XI1,XI2,FI,FI1,FI2,FX0,HI2,YXI1,YXI2,XS2,II
      1,IJ,IkJ ,CC1,CC2,CC3,CC4,CC5,DD1,DD2,DD3,DD4,DD5,DD6,DD7,DD8
      TYPE COMPLEX A,EIGG,EVAL,EVEC
      TYPE COMPLEX AE,XI,XI1,XI2,FI,FI1,FI2,HI2 ,CC1,CC2,CC3,CC4,CC5
      TYPE COMPLEX AZ,BZ,AX

      EVALUATE POLYNOMIAL ELEMENTS AT XI

      10 CALL CALFUNC(AZ)

      COMPUTE VALUE OF DETERMINANT

      20 CALL CDTERM(NN,AE,BZ,KER,20)
      25 GO TO (60,30),KER
      30 WRITE (51,40) ,IJ,II,AZ
      40 FORMAT(/4X,42HERROR IN FUNCTION EVALUATION OF EIGENVALUE,I5,2X,
      112HAT ITERATION ,I5,2X,3HXI=,C(E22.11,E22.11))
      STOP
      60 IKC=IJ-1

      DETERMINE IF APOSTERIORI SCALING IS NECESSARY

      IF(BZ-1.0E+90) 68,65,65
      65 DD8=1.0
      68 CC5=BZ
      IF (IKC) 90,90,70
      70 AX=(1.0,0.0)

      SUPPRESSION OF ROOTS PREVIOUSLY CALCULATED

      DO 80 I=1,IKC
      AX=AX*(AZ-EVAL(I)/XS2)
      80 CONTINUE
      BZ=BZ/AX
      90 RETURN
      END
003580
003590
003600
003610
003620
003630
003640
003650
003660
003670
003680
003690
003700
003710
003720
003730
003740
003750
003760
003770
003780
003790
003800
003810
003820
003830
003840
003850
003860
003870
003880
003890
003900
003910
003920
003930
003940
003950
003960
003970
003980
003990
004000

```



```

SUBROUTINE ROMAT
C
C      THIS SUBROUTINE RETURNS PROBLEM TO ORIGINAL FORM IF
C      APOSTERIORI SCALING HAS OCCURED
C
      DIMENSION AE(20,20)
      COMMON A(20,20,3),EIGG(40,2),EVAL(40),EVEC(20,40),NN,IORD,IE,NEIG,
      IIMAX,ISAC,IPRINT,IVEC,EP1,EP2,EP3
      COMMON /GBLOC2/AE,XI,XI1,XI2,FI,FI1,FI2,FX0,HI2,YXI1,YXI2,XS2,II
      1,IJ,IKJ ,CC1,CC2,CC3,CC4,CC5,DD1,DD2,DD3,DD4,DD5,DD6,DD7,DD8
      TYPE COMPLEX AE,XI,XI1,XI2,FI,FI1,FI2,HI2 ,CC1,CC2,CC3,CC4,CC5
      TYPE COMPLEX A,EIGG,EVAL,EVEC
      IF(IORD-1.0) 10,10,40
      10 DO 30 I=1,NN
      DO 20 J=1,NN
      A(I,J,1)=A(I,J,1)*XS2
      20 CONTINUE
      30 CONTINUE
      GO TO 100
      40 XS22=XS2*XS2
      DO 60 I=1,NN
      DO 50 J=1,NN
      A(I,J,1)=A(I,J,1)*XS22
      A(I,J,2)=A(I,J,2)*XS2
      50 CONTINUE
      60 CONTINUE
      100 RETURN
      END
004010
004020
004030
004040
004050
004060
004070
004080
004090
004100
004110
004120
004130
004140
004150
004160
004170
004180
004190
004200
004210
004220
004230
004240
004250
004260
004270
004280

```



```

SUBROUTINE ACCEPT
C
C      THIS SUBROUTINE DETERMINES IF PRESENT ITERATE IS ACCEPTABLE
C
C      DIMENSION AE(20,20)
COMMON A(20,20,3),EIGG(40,2),EVAL(40),EVEC(20,40),NN,IORD,IE,NEIG,
1IMAX,ISAC,IPRINT,IVEC,EP1,EP2,EP3
COMMON /GBLOC2/AE,XI,XI1,XI2,FI,FI1,FI2,FX0,HI2,YXI1,YXI2,XS2,II
1,IJ,IJ,KJ,CC1,CC2,CC3,CC4,CC5,DD1,DD2,DD3,DD4,DD5,DD6,DD7,DD8
TYPE COMPLEX AE,XI,XI1,XI2,FI,FI1,FI2,CC1,CC2,CC3,CC4,CC5
TYPE COMPLEX A,EIGG,EVAL,EVEC
IF(II-1) 10,10,20

C      INITIALIZE VARIABLES AT FIRST ITERATION
C
C      10 YXI1=CABS(XI1)
C      YXI2=CABS(XI2)
C      IKJ=1
C      20 YXI=CABS(XI)
C      30 XN=ABSF(YXI-YXI1)
C
C      CHECK FOR RELATIVE CONVERGENCE
C
C      40 IF (XN/YXI-EP1) 41,41,120
C      41 WRITE (51,45)
C      45 FORMAT(5X,34HRELATIVE CONVERGENCE CRITERION MET )
C      GO TO 110
C
C      CHECK FOR ABSOLUTE CONVERGENCE
C
C      120 IF (CABS(CC5)-EP2) 121,121,90
C      121 WRITE (51,122)
C      122 FORMAT(5X,34HABSOLUTE CONVERGENCE CRITERION MET )
C
C      SET FLAG TO STATE PROBLEM HAS CONVERGED
C
C      110 IKJ=2
C      GO TO 100
C
C      SET VALUES FOR NEXT ITERATION
C
C      90 YXI2=YXI1
C      YXI1=YXI
C      100 RETURN
C      END

```



```

C          SETUP LAST THREE VALUES TO START NEW PROBLEM
C
C
C          15 XI=XI/XS2
C          20 CALL FUNCEV(XI,FI)
C          XI1=XI1/XS2
C          30 CALL FUNCEV(XI1,FI1)
C          XI2=XI2/XS2
C          40 CALL FUNCEV(XI2,FI2)
C
C          CALCULATE NECESSARY VARIABLES TO INITIALIZE NEW PROBLEM
C
C          HI2=(FI1-FI2)/(XI1-XI2)
C          50 YXI2=CABS(XI2)
C          YXI1=CABS(XI1)
C          WRITE (51,60),XS2
C          60 FORMAT(/5X,33HEIGENVALUE SCALED, SCALE FACTOR=,E15.7)
C          RETURN
C          END

```



```

SUBROUTINE SCALE1
C
C   THIS SUBROUTINE DOES THE APRIORI SCALING
C
C
C   DIMENSION AE(20,20)
C   DIMENSION C(20)
C   COMMON A(20,20,3),EIGG(40,2),EVAL(40),EVEC(20,40),NN,IORD,IE,NEIG,
C   1IMAX,ISAC,IPRINT,IVEC,EP1,EP2,EP3
C   COMMON /GBLOC2/AE,XI,XI1,XI2,FI,FI1,FI2,FX0,HI2,YXI1,YXI2,XS2,II
C   1,IJ,IkJ ,CC1,CC2,CC3,CC4,CC5,DD1,DD2,DD3,DD4,DD5,DD6,DD7,DD8
C   TYPE COMPLEX AE,XI,XI1,XI2,FI,FI1,FI2,HI2 ,CC1,CC2,CC3,CC4,CC5
C   TYPE COMPLEX A,EIGG,EVAL,EVEC
C
C   X=(1.0,0.0)
C   DO 60 J=1,NN
C   DO 30 I=1,NN
C
C       SUM POLYNOMIAL COEFFICIENTS FOR ROW J
C
C       C(I)=CABS(A(J,I,3)+A(J,I,2)+A(J,I,1))
C   30 CONTINUE
C       XMAX=C(1)
C
C       FIND MAXIMUM ELEMENT IN ROW J
C
C       DO 40 I=2,NN
C       XMAX=MAX1F(XMAX,C(I))
C   40 CONTINUE
C
C       DETERMINE SCALE FACTOR,CHARACTERISTIC OF LOG(XMAX),BASE 2
C
C       XMAX=XMAX.AND. 377700000000000000B
C       XMAX=XMAX .OR. 40000000000000B
C
C       SCALE ROW J
C
C       DO 50 I=1,NN
C       A(J,I,3)=A(J,I,3)/XMAX
C       A(J,I,2)=A(J,I,2)/XMAX
C       A(J,I,1)=A(J,I,1)/XMAX
C   50 CONTINUE
C   60 CONTINUE
C   RETURN
C   END

```



```

C
C
C
SUBROUTINE EVEK
      THIS SUBROUTINE CALCULATES THE ASSOCIATED EIGENVECTOR

      DIMENSION AE(20,20)
      DIMENSION B(20,20),C(20,20),X(20)
      COMMON A(20,20,3),EIGG(40,2),EVAL(40),EVEC(20,40),NN,IORD,IE,NEIG,
1 IIMAX,ISAC,IPRINT,IVEC,EP1,EP2,EP3
      COMMON /GBLOC2/AE,XI,XI1,XI2,FI,F11,F12,FX0,HI2,YXI1,YXI2,XS2,II
1,IJ,IKJ,CC1,CC2,CC3,CC4,CC5,DD1,DD2,DD3,DD4,DD5,DD6,DD7,DD8
      TYPE COMPLEX A,EIGG,EVAL,EVEC
      TYPE COMPLEX AE,XI,XI1,XI2,FI,F11,F12,HI2,CC1,CC2,CC3,CC4,CC5
      TYPE COMPLEX B,C,X

      NX = NN-1

      EVALUATE FUNCTION AT EIGENVALUE IJ

      CALL CALFUNC(XI)
      IT=1
      ITL=IT
      IR=0
      GO TO 7
5  IR=IT
      IT=IT+1
      ITL=IT
7  ITT=IT-1
      IF(ITT) 10,10,50

      SET FIRST POSITION IN EIGENVECTOR=1

10 EVEC(1,IJ)=(1.0,0.0)

      REMOVE COLUMN 1 FROM MATRIX

      DO 30 IB=1,NN
      DO 20 IA=1,NX
      B(IB,IA)=AE(IB,IA+1)
20 CONTINUE
30 CONTINUE

```



```

C      PUT COLUMN 1 INTO SOLUTION VECTOR POSITION
C
C      DO 40 IB=1,NN
C      B(IB,NN)=-AE(IB,1)
C      40 CONTINUE
C      GO TO 200
C
C      50 IF(IT-NN) 100,60,60
C
C      SET LAST POSITION IN EIGENVECTOR=1
C
C      60 EVEC(NN,IJ)=(1.0,0.0)
C
C      REMOVE LAST COLUMN FROM MATRIX
C
C      DO 80 IB=1,NN
C      DO 70 IA=1,NX
C      B(IB,IA)=AE(IB,IA)
C      70 CONTINUE
C      80 CONTINUE
C
C      PUT LAST COLUMN INTO SOLUTION VECTOR POSITION
C
C      DO 90 IB=1,NN
C      B(IB,NN)=-AE(IB,NN)
C      90 CONTINUE
C      GO TO 200
C
C      SET POSITION IT OF EIGENVECTOR=1
C
C      100 EVEC(IT,IJ)=(1.0,0.0)
C
C      REMOVE COLUMN IT FROM MATRIX
C
C      DO 130 IB=1,NN
C      DO 120 IA=1,ITT
C      B(IB,IA)=AE(IB,IA)
C      120 CONTINUE
C      130 CONTINUE
C      DO 150 IB=1,NN
C      DO 140 IA=IT,NX
C      B(IB,IA)=AE(IB,IA+1)
C      140 CONTINUE
C      150 CONTINUE

```



```

C      PUT COLUMN IT INTO SOLUTION VECTOR POSITION
C
C      DO 160 IB=1,NN
C      B(IB,NN)=-AE(IB,IT)
C      160 CONTINUE
C      200 IRR=IR
C      IR=IR+1
C      IF (IR-NN) 205,205,202
C      202 IR=1
C      IRR=0
C      ITL=-IT
C      205 IF (IRR) 210,210,240
C
C      REMOVE FIRST ROW TO COMPLETE SETUP OF REDUCED MATRIX
C
C      210 DO 230 IB=1,NX
C      DO 220 IA=1,NN
C      C(IB,IA)=B(IB+1,IA)
C      220 CONTINUE
C      230 CONTINUE
C      GO TO 350
C      240 IF(IR-NN) 280,250,250
C
C      REMOVE LAST ROW TO COMPLETE SETUP OF REDUCED MATRIX
C
C      250 DO 270 IB=1,NX
C      DO 260 IA=1,NN
C      C(IB,IA)=B(IB,IA)
C      260 CONTINUE
C      270 CONTINUE
C      GO TO 350
C
C      REMOVE ROW IR TO COMPLETE SETUP OF REDUCED MATRIX
C
C      280 DO 300 IB=1,IRR
C      DO 290 IA=1,NN
C      C(IB,IA)=B(IB,IA)
C      290 CONTINUE
C      300 CONTINUE

```

```

006630
006640
006650
006660
006670
006680
006690
006700
006710
006720
006730
006740
006750
006760
006770
006780
006790
006800
006810
006820
006830
006840
006850
006860
006870
006880
006890
006900
006910
006920
006930
006940
006950
006960
006970
006980
006990
007000
007010
007020

```



```

DO 320 IB=IR,NX
DO 310 IA=1,NN
C(IB,IA)=B(IB+1,IA)
310 CONTINUE
320 CONTINUE
C
C      SOLVE REDUCED SYSTEM
C
350 CALL JORCOM(C,NX,X,20,20)
C
C      PLACE SOLUTION OF REDUCED SYSTEM INTO EIGENVECTOR
C
IF(ITT) 360,360,380
360 DO 370 IA=2,NN
EVEC(IA,IJ)=X(IA-1)
370 CONTINUE
GO TO 450
380 IF(IT-NN) 410,390,390
390 DO 400 IA=1,NX
EVEC(IA,IJ)=X(IA)
400 CONTINUE
GO TO 450
410 DO 420 IB=1,ITT
EVEC(IB,IJ)=X(IB)
420 CONTINUE
DO 430 IB=IT,NX
EVEC(IB+1,IJ)=X(IB)
430 CONTINUE
C
C      CHECK SOLUTION TO DETERMINE ACCEPTABILITY
C
450 CALL CKEVEC
IF(DD2) 470,455,470
455 WRITE (51,460) ,(EVEC(IB,IJ),IB=1,NN)
460 FORMAT(/4X,23HASSOCIATED EIGENVECTOR ,/(C(E22.11, E22.11)))
GO TO 500
470 IF (IR-ITL-1) 200,5,480
480 WRITE (51,490) , (EVEC(IB,IJ),IB=1,NN)
490 FORMAT(/4X,38HUNABLE TO FIND EIGENVECTOR, LAST TRY= ,/(C(E22.11,E
122.11)))
500 RETURN
END

```


APPENDIX III

A. IDENTIFICATION:

Title: A Method for the Solution of the "Generalized Eigenvalue Problem" with Polynomial Matrix Elements.

CO-OP ID: F4-NPGS-GENEIG (1604 F-63)

Category: Mathematical Subroutine

Programmer: R. D. Brunell

Date: June 10, 1966

B. PURPOSE:

To find those values of the complex parameter, z , which satisfy the system of equations:

$$A_{11}(z)x_1 + A_{12}(z)x_2 + \dots + A_{1n}(z)x_n = 0$$

$$A_{21}(z)x_1 + A_{22}(z)x_2 + \dots + A_{2n}(z)x_n = 0$$

$$(1) \quad \begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} \quad \begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} \quad \begin{array}{c} A_{n1}(z)x_1 + A_{n2}(z)x_2 + \dots + A_{nn}(z)x_n = 0, \end{array}$$

where the a_{ij} 's are complex polynomials of degree 2 in z .

This may be written more concisely as

$$(2) \quad H(z)X = 0,$$

where $H(z)$ is an $n \times n$ matrix whose elements are the polynomials in z , and X is a complex column vector of n elements. The elements of the set $\{z_i: H(z_i)X_i = 0\}$ are called the eigenvalues, and the X_i are called the associated eigenvectors.

The matrix $H(z)$, because of the form of its elements, may be written in the following form:

$$(3) \quad H(z) = A_0 z^2 + A_1 z + A_2,$$

where the $n \times n$ matrices A_i have constant complex elements.

Equation (3) will be useful later when discussing the input for the subroutine.

C. USAGE:

1. Set up for GENEIG

GENEIG is a subroutine and the user must supply a main program which will set up the input data for GENEIG. All communication to and from this subroutine is done through common storage, and the following DIMENSION, COMMON and TYPE statements must be declared by the user.

DIMENSION A(20,20,3),EIGG(40,2),EVAL(40),EVEC(20,40)

COMMON A,EIGG,EVAL,EVEC,NN,IORD,IE,
 NEIG,IMAX,ISAC,IPRINT,IVEC,EPI,EP2,EP3

TYPE COMPLEX A,EIGG,EVAL,EVEC

2. Input for GENEIG

The three-dimensional array, A, must contain the elements of the matrix H in the manner described below:

$$a_{ij}^{(0)} = A(I,J,1), \quad a_{ij}^{(1)} = A(I,J,2) \quad \text{and} \quad a_{ij}^{(2)} = A(I,J,3),$$

where $a_{ij}^{(k)}$ is the ij -th element of A_k .

The two-dimensional array, EIGG, must contain the estimated value for the eigenvalues as follows:

EIGG(I,1) will be considered as the estimate to the value of the i -th eigenvalue, and EIGG(I,2) will be considered as the probable percentage error in that guess. The first three values necessary to start the process for the i -th value are computed by

$$z_i^{(1)} = \text{EIGG}(I,1) - \text{EIGG}(I,2)*\text{EIGG}(I,1)$$

$$z_i^{(2)} = \text{EIGG}(I,1)$$

(4)

$$z_i^{(3)} = \text{EIGG}(I,1) + \text{EIGG}(I,2)*\text{EIGG}(I,1) \quad .$$

The array EVEC will contain the associated eigenvectors. The eigenvector associated with the i -th eigenvalue will be found in EVEC(J,I), where $J = 1,2,\dots,NN$.

NN is the order of the coefficient matrices.

IORD is the degree of the polynomial elements.

IE is the number of the first eigenvalue to be found.

NEIG is the number of the last eigenvalue to be found.

IMAX is the maximum number of iterations to be allowed.

ISAC is a flag for initial scaling. If ISAC = 0 no scaling will occur.

IPRINT is a flag for printing at every iteration. If IPRINT = 0 only the final results will be printed.

IVEC is a flag for the eigenvector option. If IVEC = 0 no eigenvectors will be computed.

EPI is the relative error bound on the eigenvalues. If

$|z_i - z_{i-1}| / |z_i| < \text{EPI}$ the problem is considered to have converged.

A value of $10^{-4} \leq \text{EPI} \leq 10^{-8}$ has been found to be satisfactory for most cases.

EP2 is an absolute error bound on the function. If $|H(z_i)| < \text{EP2}$

the problem is considered to have converged. A value of $\text{EP2} \approx 10^{-15}$ appears to be satisfactory for those cases for which this test becomes effective.

EP3 is a relative error bound on the eigenvector. If B is the maximum

element of $H(z_i)$, then if $|H(z_i)x_i| < |\text{EP3} \cdot B|$ the eigenvector is considered acceptable. A value of $\text{EP3} \approx 10^{-4}$ is satisfactory.

3. Calling Sequence:

Once the common storage has been set up in the user's program it is only necessary to insert the instruction CALL GENEIG. The subroutine will then compute all eigenvalues and eigenvectors, that have been indicated by the input, before returning control to the calling program.

4. Space Required: 4217_{10} excluding common storage
5. Common Storage Required: 4251_{10}
6. Print-Outs: If IPRINT = 0 then at each iteration the number of the iteration, the present value of z_i and $H(z_i)$ will be printed.

When an eigenvalue is accepted, a message to this effect will be printed along with the accepted value and the function value.

If IVEC = 0 and an eigenvector is found it will be printed. However, if no acceptable eigenvector is found, a statement to this effect will be printed along with the value of the final attempt.

7. Error Stops: If IORD is found to be other than an integer 1 or 2 the program will print the value of IORD and stop (Exit to Monitor). If the determinant evaluator is unable to complete its operation the code will indicate that this has happened and stop (Exit to Monitor).
8. Input and Output Formats: Not applicable.
9. Input and Output Tape Mountings: Output is to Logical unit 51 (Standard on CO-OP Monitor).
10. Selective Jump and Stop Settings: Not applicable.
11. Timing Examples:

Case	1	2	3	4	5	6
Size of Matrix	3x3	4x4	5x5	8x8	12x12	16x16
No. of Eigenvalues	6	4	5	8	12	4
No. of Eigenvectors	6	4	0	0	0	0
Average No. of Iterations	4.0	8.25	6.0	10.5	18.7	17.5
Total Time (seconds)	3.85	3.85	4.38	16.0	104.	55.28

12. Accuracy:

Results from this code have been compared with results taken from several papers on the standard eigenvalue problem and they have been as accurate or more accurate on all tests. Also, several problems were run for which exact results were known. On problems with well-defined roots nine or ten figure accuracy was observed. When pathological roots were encountered the accuracy of this code compared very favorably with other methods tested.

13. Cautions to User: The DIMENSION, COMMON and TYPE statements as shown in Section C must be used exactly as indicated.
14. Equipment Configuration: Standard CO-OP Computer with FORTRAN 63 Compiler and CO-OP Monitor System.
15. References:
 - (a) R. D. Brunell, An Iterative Solution to the Generalized Eigenvalue-Eigenvector Problem, Technical Report/Research Paper No. 69, U. S. Naval Postgraduate School, Monterey, California, 1966.
 - (b) J. F. Traub, Iterative Methods for the Solution of Equations, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1964.
 - (c) I. Tarnove, "Determination of Eigenvalues of Matrices Having Polynomial Elements," Journal of SIAM, June 1958.

D. METHOD:

Only a general discussion of the method will be presented here. For a more detailed description see Reference (a).

The heart of the method used in this subroutine is the iterative root-finding technique, which was developed by J. F. Traub (b). In essence the technique is to perform a quadratic fit using the last three iterants and their functional values. The new estimate to the root is defined as that root of the fitting quadratic which is closer to the last iterant. The process converges with almost arbitrary estimates; however, through the use of good estimates a considerable saving in computer time can be realized. Fewer iterations will be necessary and only those eigenvalues of interest need be computed.

In order to avoid convergence to a previously-calculated root a method of suppressing zeros, first suggested by G. E. Forsythe (Reference (c) page 164), was incorporated into the code. This feature can also be used to eliminate re-calculation of roots found on a previous run, especially when they may lie pathologically close to a desired root. This can be accomplished by loading the m known roots into the first m positions in the EVAL array and setting IE equal to m + 1.

To alleviate the problem of floating-point exponential overflow a two-part, a priori and a posteriori, automatic scaling procedure was devised for this code.

The a priori scaling is optional and is controlled by the flag ISAC. This scaling occurs only once and is done prior to any calculations for eigenvalues. The coefficients of each polynomial element are summed and a maximum, B , is found in each row. The scaling is carried out by dividing each coefficient of every polynomial in a row by 2^{J+1} , where $2^j \leq B \leq 2^{J+1}$.

The a posteriori scaling is not optional and will occur only when an overflow is sensed. At this point the problem is transformed so that the scaled eigenvalue will be exactly the original eigenvalue divided by 2^{k+1} , where $2^k \leq z_i \leq 2^{k+1}$ and z_i is the value of the iterant at the time of the overflow.

The computation of the associated eigenvectors is an optional feature in this code. The basic assumption in this section is that the eigenvalue reduces the rank of the original matrix by 1. Under this assumption 2NN attempts will be made to find an acceptable eigenvector. If all attempts fail, a printout to this effect will be made.

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) U. S. Naval Postgraduate School Monterey, California		2 a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	
		2 b. GROUP	
3. REPORT TITLE AN ITERATIVE SOLUTION TO THE GENERALIZED EIGENVALUE-EIGENVECTOR PROBLEM			
4. DESCRIPTIVE NOTES (Type of report and inclusive dates)			
5. AUTHOR(S) (Last name, first name, initial) Brunell, Ronald D.			
6. REPORT DATE 12 July 1966		7 a. TOTAL NO. OF PAGES 52	7 b. NO. OF REFS
8 a. CONTRACT OR GRANT NO.		9 a. ORIGINATOR'S REPORT NUMBER(S) TR/RP No. 69	
b. PROJECT NO.		9 b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)	
c.			
d.			
10. AVAILABILITY/LIMITATION NOTICES Unlimited			
11. SUPPLEMENTARY NOTES		12. SPONSORING MILITARY ACTIVITY	
13. ABSTRACT The abstract is as given on the title page.			

KEY WORDS

LINK A

LINK B

LINK C

ROLE

WT

ROLE

WT

ROLE

WT

Generalized

Eigenvalue

Eigenvector

INSTRUCTIONS

ORIGINATING ACTIVITY: Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization (*corporate author*) issuing the report.

REPORT SECURITY CLASSIFICATION: Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

GROUP: Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

REPORT TITLE: Enter the complete report title in all capital letters. Titles in all cases should be unclassified. A meaningful title cannot be selected without classification. Show title classification in all capitals in parenthesis immediately following the title.

DESCRIPTIVE NOTES: If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

AUTHOR(S): Enter the name(s) of author(s) as shown on the report. Enter last name, first name, middle initial. Military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

REPORT DATE: Enter the date of the report as day, month, year, or month, year. If more than one date appears on the report, use date of publication.

TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

NUMBER OF REFERENCES: Enter the total number of references cited in the report.

CONTRACT OR GRANT NUMBER: If appropriate, enter the applicable number of the contract or grant under which the report was written.

PROJECT NUMBER: Enter the appropriate military department identification, such as project number, project number, system numbers, task number, etc.

ORIGINATOR'S REPORT NUMBER(S): Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

OTHER REPORT NUMBER(S): If the report has been assigned any other report numbers (either by the originator or by the sponsor), also enter this number(s).

AVAILABILITY/LIMITATION NOTICES: Enter any limitations on further dissemination of the report, other than those

imposed by security classification, using standard statements such as:

- (1) "Qualified requesters may obtain copies of this report from DDC."
- (2) "Foreign announcement and dissemination of this report by DDC is not authorized."
- (3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through _____."
- (4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through _____."
- (5) "All distribution of this report is controlled. Qualified DDC users shall request through _____."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING MILITARY ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring (paying for) the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as (TS), (S), (C), or (U).

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, roles, and weights is optional.

DISTRIBUTION LIST

Documents Department
General Library
University of California
Berkeley, California 94720

Lockheed-California Company
Central Library
Dept. 77-14, Bldg. 170, Plt. B-1
Burbank, California 91503

Naval Ordnance Test Station
China Lake, California
Attn: Technical Library

Serials Dept., Library
University of California, San Diego
La Jolla, California 92038

Aircraft Division
Douglas Aircraft Company, Inc.
3855 Lakewood Boulevard
Long Beach, California 90801
Attn: Technical Library

Librarian
Government Publications Room
University of California
Los Angeles, California 90024

Librarian
Numerical Analysis Research
University of California
405 Hilgard Avenue
Los Angeles, California 90024

Chief Scientist
Office of Naval Research
Branch Office
1030 East Green Street
Pasadena, California 91101

Commanding Officer and Director
U. S. Navy Electronics Lab. (Library)
San Diego, California 92152

General Dynamics/Convair
P.O. Box 1950
San Diego, California 92112
Attn: Engineering Library
Mail Zone 6-157

Ryan Aeronautical Company
Attn: Technical Information
Services
Lindbergh Field
San Diego, California 92112

General Electric Company
Technical Information Center
P.O. Drawer QQ
Santa Barbara, California 93102

Library
Boulder Laboratories
National Bureau of Standards
Boulder, Colorado 80302

Government Documents Division
University of Colorado Libraries
Boulder, Colorado 80304

The Library
United Aircraft Corporation
400 Main Street
East Hartford, Connecticut 06108

Documents Division
Yale University Library
New Haven, Connecticut 06520

Librarian
Bureau of Naval Weapons
Washington, D. C. 20360

George Washington University Library
2023 G Street, N. W.
Washington, D. C. 20006

National Bureau of Standards Library
Room 301, Northwest Building
Washington, D. C. 20234

Director
Naval Research Laboratory
Washington, D. C. 20390
Attn: Code 2027

University of Chicago Library
Serial Records Department
Chicago, Illinois 60637

Documents Department
Northwestern University Library
Evanston, Illinois 60201

The Technological Institute, Library
Northwestern University
Evanston, Illinois 60201

Librarian
Purdue University
Lafayette, Indiana 47907

Johns Hopkins University Library
Baltimore
Maryland 21218

Martin Company
Science-Technology Library
Mail 398
Baltimore, Maryland 21203

Scientific and Technical Information
Facility
Attn: NASA Representative
P.O. Box 5700
Bethesda, Maryland 20014

Documents Office
University of Maryland Library
College Park, Maryland 20742

The Johns Hopkins University
Applied Physics Laboratory
Silver Spring, Maryland
Attn: Document Librarian

Librarian
Technical Library, Code 245L
Building 39/3
Boston Naval Shipyard
Boston, Massachusetts 02129

Massachusetts Institute of Technology
Serials and Documents
Hayden Library
Cambridge, Massachusetts 02139

Technical Report Collection
303A, Pierce Hall
Harvard University
Cambridge, Massachusetts 02138
Attn: Mr. John A. Harrison, Librarian

Alumni Memorial Library
Lowell Technological Institute
Lowell, Massachusetts

Librarian
University of Michigan
Ann Arbor, Michigan 48104

Gifts and Exchange Division
Walter Library
University of Minnesota
Minneapolis, Minnesota 55455

Reference Department
John M. Olin Library
Washington University
6600 Millbrook Boulevard
St. Louis, Missouri 63130

Librarian
Forrestal Research Center
Princeton University
Princeton, New Jersey 08540

U. S. Naval Air Turbine Test Station
Attn: Foundational Research Coordinator
Trenton, New Jersey 08607

Engineering Library
Plant 25
Grumman Aircraft Engineering Corp.
Bethpage, L. I., New York 11714

Librarian
Fordham University
Bronx, New York 10458

U. S. Naval Applied Science Laboratory
Technical Library
Building 291, Code 9832
Naval Base
Brooklyn, New York 11251

Librarian
Cornell Aeronautical Laboratory
4455 Genesee Street
Buffalo, New York 14225

Central Serial Record Dept.
Cornell University Library
Ithaca, New York 14850

Columbia University Libraries
Documents Acquisitions
535 W. 114 Street
New York, New York 10027

Engineering Societies Library
345 East 47th Street
New York, New York 10017

Library-Serials Department
Rensselaer Polytechnic Institute
Troy, New York 12181

Librarian
Documents Division
Duke University
Durham, North Carolina 27706

Ohio State University Libraries
Serial Division
1858 Neil Avenue
Columbus, Ohio 43210

Commander
Philadelphia Naval Shipyard
Philadelphia, Pennsylvania 19112
Attn: Librarian, Code 249c

Steam Engineering Library
Westinghouse Electric Corporation
Lester Branch Postoffice
Philadelphia, Pennsylvania 19113

Hunt Library
Carnegie Institute of Technology
Pittsburgh, Pennsylvania 15213

Documents Division
Brown University Library
Providence, Rhode Island 02912

Central Research Library
Oak Ridge National Laboratory
Post Office Box X
Oak Ridge, Tennessee 37831

Documents Division
The Library
Texas A & M University
College Station, Texas 77843

Librarian
LTV Vought Aeronautics Division
P.O. Box 5907
Dallas, Texas 75222

Gifts and Exchange Section
Periodicals Department
University of Utah Libraries
Salt Lake City, Utah 84112

Defense Documentation Center (DDC)
Cameron Station
Alexandria, Virginia 22314
Attn: IRS (20 copies)

FOREIGN COUNTRIES

Engineering Library
Hawker Siddeley Engineering
Box 6001
Toronto International Airport
Ontario, Canada
Attn: Mrs. M. News, Librarian

Exchange Section
National Lending Library for
Science and Technology
Boston Spa
Yorkshire, England

The Librarian
Patent Office Library
25 Southampton Buildings
Chancery Lane
London W. C. 2., England

Librarian
National Inst. of Oceanography
Wormley, Godalming
Surrey, England

Dr. H. Tigerschiöld, Director
Library
Chalmers University of Technology
Gibraltargatan 5
Gothenburg S, Sweden

International Computation Centre
Palazzo degli Uffici-Zona dell' E.U.R.
Rome, Italy

TA7

.U62 Brunell

no.69 An iterative solution
to the generalized
eigenvalue-eigenvector
problem.

98302

genTA 7.U62 no.69

An iterative solution to the generalized



3 2768 001 61422 5

DUDLEY KNOX LIBRARY

1911-1912

<p> Date </p>	<p> Description </p>	<p> Amount </p>	<p> Balance </p>	<p> Total </p>	<p> Remarks </p>
<p> Jan 1 </p>	<p> Balance </p>	<p> 100.00 </p>	<p> 100.00 </p>	<p> 100.00 </p>	<p> Balance </p>
<p> Jan 2 </p>	<p> Cash </p>	<p> 50.00 </p>	<p> 150.00 </p>	<p> 150.00 </p>	<p> Cash </p>
<p> Jan 3 </p>	<p> Cash </p>	<p> 25.00 </p>	<p> 175.00 </p>	<p> 175.00 </p>	<p> Cash </p>
<p> Jan 4 </p>	<p> Cash </p>	<p> 75.00 </p>	<p> 250.00 </p>	<p> 250.00 </p>	<p> Cash </p>
<p> Jan 5 </p>	<p> Cash </p>	<p> 100.00 </p>	<p> 350.00 </p>	<p> 350.00 </p>	<p> Cash </p>
<p> Jan 6 </p>	<p> Cash </p>	<p> 50.00 </p>	<p> 400.00 </p>	<p> 400.00 </p>	<p> Cash </p>
<p> Jan 7 </p>	<p> Cash </p>	<p> 25.00 </p>	<p> 425.00 </p>	<p> 425.00 </p>	<p> Cash </p>
<p> Jan 8 </p>	<p> Cash </p>	<p> 75.00 </p>	<p> 500.00 </p>	<p> 500.00 </p>	<p> Cash </p>
<p> Jan 9 </p>	<p> Cash </p>	<p> 100.00 </p>	<p> 600.00 </p>	<p> 600.00 </p>	<p> Cash </p>
<p> Jan 10 </p>	<p> Cash </p>	<p> 50.00 </p>	<p> 650.00 </p>	<p> 650.00 </p>	<p> Cash </p>
<p> Jan 11 </p>	<p> Cash </p>	<p> 25.00 </p>	<p> 675.00 </p>	<p> 675.00 </p>	<p> Cash </p>
<p> Jan 12 </p>	<p> Cash </p>	<p> 75.00 </p>	<p> 750.00 </p>	<p> 750.00 </p>	<p> Cash </p>
<p> Jan 13 </p>	<p> Cash </p>	<p> 100.00 </p>	<p> 850.00 </p>	<p> 850.00 </p>	<p> Cash </p>
<p> Jan 14 </p>	<p> Cash </p>	<p> 50.00 </p>	<p> 900.00 </p>	<p> 900.00 </p>	<p> Cash </p>
<p> Jan 15 </p>	<p> Cash </p>	<p> 25.00 </p>	<p> 925.00 </p>	<p> 925.00 </p>	<p> Cash </p>
<p> Jan 16 </p>	<p> Cash </p>	<p> 75.00 </p>	<p> 1000.00 </p>	<p> 1000.00 </p>	<p> Cash </p>
<p> Jan 17 </p>	<p> Cash </p>	<p> 100.00 </p>	<p> 1100.00 </p>	<p> 1100.00 </p>	<p> Cash </p>
<p> Jan 18 </p>	<p> Cash </p>	<p> 50.00 </p>	<p> 1150.00 </p>	<p> 1150.00 </p>	<p> Cash </p>
<p> Jan 19 </p>	<p> Cash </p>	<p> 25.00 </p>	<p> 1175.00 </p>	<p> 1175.00 </p>	<p> Cash </p>
<p> Jan 20 </p>	<p> Cash </p>	<p> 75.00 </p>	<p> 1250.00 </p>	<p> 1250.00 </p>	<p> Cash </p>
<p> Jan 21 </p>	<p> Cash </p>	<p> 100.00 </p>	<p> 1350.00 </p>	<p> 1350.00 </p>	<p> Cash </p>
<p> Jan 22 </p>	<p> Cash </p>	<p> 50.00 </p>	<p> 1400.00 </p>	<p> 1400.00 </p>	<p> Cash </p>
<p> Jan 23 </p>	<p> Cash </p>	<p> 25.00 </p>	<p> 1425.00 </p>	<p> 1425.00 </p>	<p> Cash </p>
<p> Jan 24 </p>	<p> Cash </p>	<p> 75.00 </p>	<p> 1500.00 </p>	<p> 1500.00 </p>	<p> Cash </p>
<p> Jan 25 </p>	<p> Cash </p>	<p> 100.00 </p>	<p> 1600.00 </p>	<p> 1600.00 </p>	<p> Cash </p>
<p> Jan 26 </p>	<p> Cash </p>	<p> 50.00 </p>	<p> 1650.00 </p>	<p> 1650.00 </p>	<p> Cash </p>
<p> Jan 27 </p>	<p> Cash </p>	<p> 25.00 </p>	<p> 1675.00 </p>	<p> 1675.00 </p>	<p> Cash </p>
<p> Jan 28 </p>	<p> Cash </p>	<p> 75.00 </p>	<p> 1750.00 </p>	<p> 1750.00 </p>	<p> Cash </p>
<p> Jan 29 </p>	<p> Cash </p>	<p> 100.00 </p>	<p> 1850.00 </p>	<p> 1850.00 </p>	<p> Cash </p>
<p> Jan 30 </p>	<p> Cash </p>	<p> 50.00 </p>	<p> 1900.00 </p>	<p> 1900.00 </p>	<p> Cash </p>
<p> Jan 31 </p>	<p> Cash </p>	<p> 25.00 </p>	<p> 1925.00 </p>	<p> 1925.00 </p>	<p> Cash </p>
<p> Feb 1 </p>	<p> Cash </p>	<p> 75.00 </p>	<p> 2000.00 </p>	<p> 2000.00 </p>	<p> Cash </p>
<p> Feb 2 </p>	<p> Cash </p>	<p> 100.00 </p>	<p> 2100.00 </p>	<p> 2100.00 </p>	<p> Cash </p>
<p> Feb 3 </p>	<p> Cash </p>	<p> 50.00 </p>	<p> 2150.00 </p>	<p> 2150.00 </p>	<p> Cash </p>
<p> Feb 4 </p>	<p> Cash </p>	<p> 25.00 </p>	<p> 2175.00 </p>	<p> 2175.00 </p>	<p> Cash </p>
<p> Feb 5 </p>	<p> Cash </p>	<p> 75.00 </p>	<p> 2250.00 </p>	<p> 2250.00 </p>	<p> Cash </p>
<p> Feb 6 </p>	<p> Cash </p>	<p> 100.00 </p>	<p> 2350.00 </p>	<p> 2350.00 </p>	<p> Cash </p>
<p> Feb 7 </p>	<p> Cash </p>	<p> 50.00 </p>	<p> 2400.00 </p>	<p> 2400.00 </p>	<p> Cash </p>
<p> Feb 8 </p>	<p> Cash </p>	<p> 25.00 </p>	<p> 2425.00 </p>	<p> 2425.00 </p>	<p> Cash </p>
<p> Feb 9 </p>	<p> Cash </p>	<p> 75.00 </p>	<p> 2500.00 </p>	<p> 2500.00 </p>	<p> Cash </p>
<p> Feb 10 </p>	<p> Cash </p>	<p> 100.00 </p>	<p> 2600.00 </p>	<p> 2600.00 </p>	<p> Cash </p>
<p> Feb 11 </p>	<p> Cash </p>	<p> 50.00 </p>	<p> 2650.00 </p>	<p> 2650.00 </p>	<p> Cash </p>
<p> Feb 12 </p>	<p> Cash </p>	<p> 25.00 </p>	<p> 2675.00 </p>	<p> 2675.00 </p>	<p> Cash </p>
<p> Feb 13 </p>	<p> Cash </p>	<p> 75.00 </p>	<p> 2750.00 </p>	<p> 2750.00 </p>	<p> Cash </p>
<p> Feb 14 </p>	<p> Cash </p>	<p> 100.00 </p>	<p> 2850.00 </p>	<p> 2850.00 </p>	<p> Cash </p>
<p> Feb 15 </p>	<p> Cash </p>	<p> 50.00 </p>	<p> 2900.00 </p>	<p> 2900.00 </p>	<p> Cash </p>
<p> Feb 16 </p>	<p> Cash </p>	<p> 25.00 </p>	<p> 2925.00 </p>	<p> 2925.00 </p>	<p> Cash </p>
<p> Feb 17 </p>	<p> Cash </p>	<p> 75.00 </p>	<p> 3000.00 </p>	<p> 3000.00 </p>	<p> Cash </p>
<p> Feb 18 </p>	<p> Cash </p>	<p> 100.00 </p>	<p> 3100.00 </p>	<p> 3100.00 </p>	<p> Cash </p>
<p> Feb 19 </p>	<p> Cash </p>	<p> 50.00 </p>	<p> 3150.00 </p>	<p> 3150.00 </p>	<p> Cash </p>
<p> Feb 20 </p>	<p> Cash </p>	<p> 25.00 </p>	<p> 3175.00 </p>	<p> 3175.00 </p>	<p> Cash </p>
<p> Feb 21 </p>	<p> Cash </p>	<p> 75.00 </p>	<p> 3250.00 </p>	<p> 3250.00 </p>	<p> Cash </p>
<p> Feb 22 </p>	<p> Cash </p>	<p> 100.00 </p>	<p> 3350.00 </p>	<p> 3350.00 </p>	<p> Cash </p>
<p> Feb 23 </p>	<p> Cash </p>	<p> 50.00 </p>	<p> 3400.00 </p>	<p> 3400.00 </p>	<p> Cash </p>
<p> Feb 24 </p>	<p> Cash </p>	<p> 25.00 </p>	<p> 3425.00 </p>	<p> 3425.00 </p>	<p> Cash </p>
<p> Feb 25 </p>	<p> Cash </p>	<p> 75.00 </p>	<p> 3500.00 </p>	<p> 3500.00 </p>	<p> Cash </p>
<p> Feb 26 </p>	<p> Cash </p>	<p> 100.00 </p>	<p> 3600.00 </p>	<p> 3600.00 </p>	<p> Cash </p>
<p> Feb 27 </p>	<p> Cash </p>	<p> 50.00 </p>	<p> 3650.00 </p>	<p> 3650.00 </p>	<p> Cash </p>
<p> Feb 28 </p>	<p> Cash </p>	<p> 25.00 </p>	<p> 3675.00 </p>	<p> 3675.00 </p>	<p> Cash </p>
<p> Feb 29 </p>	<p> Cash </p>	<p> 75.00 </p>	<p> 3750.00 </p>	<p> 3750.00 </p>	<p> Cash </p>
<p> Mar 1 </p>	<p> Cash </p>	<p> 100.00 </p>	<p> 3850.00 </p>	<p> 3850.00 </p>	<p> Cash </p>
<p> Mar 2 </p>	<p> Cash </p>	<p> 50.00 </p>	<p> 3900.00 </p>	<p> 3900.00 </p>	<p> Cash </p>
<p> Mar 3 </p>	<p> Cash </p>	<p> 25.00 </p>	<p> 3925.00 </p>	<p> 3925.00 </p>	<p> Cash </p>
<p> Mar 4 </p>	<p> Cash </p>	<p> 75.00 </p>	<p> 4000.00 </p>	<p> 4000.00 </p>	<p> Cash </p>
<p> Mar 5 </p>	<p> Cash </p>	<p> 100.00 </p>	<p> 4100.00 </p>	<p> 4100.00 </p>	<p> Cash </p>
<p> Mar 6 </p>	<p> Cash </p>	<p> 50.00 </p>	<p> 4150.00 </p>	<p> 4150.00 </p>	<p> Cash </p>
<p> Mar 7 </p>	<p> Cash </p>	<p> 25.00 </p>	<p> 4175.00 </p>	<p> 4175.00 </p>	<p> Cash </p>
<p> Mar 8 </p>	<p> Cash </p>	<p> 75.00 </p>	<p> 4250.00 </p>	<p> 4250.00 </p>	<p> Cash </p>
<p> Mar 9 </p>	<p> Cash </p>	<p> 100.00 </p>	<p> 4350.00 </p>	<p> 4350.00 </p>	<p> Cash </p>
<p> Mar 10 </p>	<p> Cash </p>	<p> 50.00 </p>	<p> 4400.00 </p>	<p> 4400.00 </p>	<p> Cash </p>
<p> Mar 11 </p>	<p> Cash </p>	<p> 25.00 </p>	<p> 4425.00 </p>	<p> 4425.00 </p>	<p> Cash </p>
<p> Mar 12 </p>	<p> Cash </p>	<p> 75.00 </p>	<p> 4500.00 </p>	<p> 4500.00 </p>	<p> Cash </p>
<p> Mar 13 </p>	<p> Cash </p>	<p> 100.00 </p>	<p> 4600.00 </p>	<p> 4600.00 </p>	<p> Cash </p>
<p> Mar 14 </p>	<p> Cash </p>	<p> 50.00 </p>	<p> 4650.00 </p>	<p> 4650.00 </p>	<p> Cash </p>
<p> Mar 15 </p>	<p> Cash </p>	<p> 25.00 </p>	<p> 4675.00 </p>	<p> 4675.00 </p>	<p> Cash </p>
<p> Mar 16 </p>	<p> Cash </p>	<p> 75.00 </p>	<p> 4750.00 </p>	<p> 4750.00 </p>	<p> Cash </p>
<p> Mar 17 </p>	<p> Cash </p>	<p> 100.00 </p>	<p> 4850.00 </p>	<p> 4850.00 </p>	<p> Cash </p>
<p> Mar 18 </p>	<p> Cash </p>	<p> 50.00 </p>	<p> 4900.00 </p>	<p> 4900.00 </p>	<p> Cash </p>
<p> Mar 19 </p>	<p> Cash </p>	<p> 25.00 </p>	<p> 4925.00 </p>	<p> 4925.00 </p>	<p> Cash </p>
<p> Mar 20 </p>	<p> Cash </p>	<p> 75.00 </p>	<p> 5000.00 </p>	<p> 5000.00 </p>	<p> Cash </p>
<p> Mar 21 </p>	<p> Cash </p>	<p> 100.00 </p>	<p> 5100.00 </p>	<p> 5100.00 </p>	<p> Cash </p>
<p> Mar 22 </p>	<p> Cash </p>	<p> 50.00 </p>	<p> 5150.00 </p>	<p> 5150.00 </p>	<p> Cash </p>
<p> Mar 23 </p>	<p> Cash </p>	<p> 25.00 </p>	<p> 5175.00 </p>	<p> 5175.00 </p>	<p> Cash </p>
<p> Mar 24 </p>	<p> Cash </p>	<p> 75.00 </p>	<p> 5250.00 </p>	<p> 5250.00 </p>	<p> Cash </p>
<p> Mar 25 </p>	<p> Cash </p>	<p> 100.00 </p>	<p> 5350.00 </p>	<p> 5350.00 </p>	<p> Cash </p>
<p> Mar 26 </p>	<p> Cash </p>	<p> 50.00 </p>	<p> 5400.00 </p>	<p> 5400.00 </p>	<p> Cash </p>
<p> Mar 27 </p>	<p> Cash </p>	<p> 25.00 </p>	<p> 5425.00 </p>	<p> 5425.00 </p>	<p> Cash </p>
<p> Mar 28 </p>	<p> Cash </p>	<p> 75.00 </p>	<p> 5500.00 </p>	<p> 5500.00 </p>	<p> Cash </p>
<p> Mar 29 </p>	<p> Cash </p>	<p> 100.00 </p>	<p> 5600.00 </p>	<p> 5600.00 </p>	<p> Cash </p>
<p> Mar 30 </p>	<p> Cash </p>	<p> 50.00 </p>	<p> 5650.00 </p>	<p> 5650.00 </p>	<p> Cash </p>
<p> Mar 31 </p>	<p> Cash </p>	<p> 25.00 </p>	<p> 5675.00 </p>	<p> 5675.00 </p>	<p> Cash </p>